Université Paris-Dauphine
Département MIDO

M2, Analyse & Probabilités.

T.P. 4 : Finite Elements

1 With Matlab/Octave

We start by implementing in Matlab/Octave

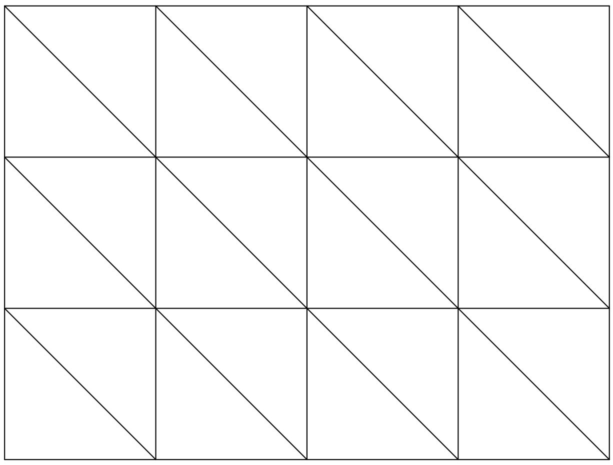
1.1 In 1D

Consider $\Omega = [0, 1]$.

1. Compute the mass and Stiffness matrices. Solve
$$\frac{u^{n+1} - u^n}{dt} = \Delta u^{n+1} + f,$$
which is equivalent to
$$u^{n+1} - dt\Delta u^{n+1} = u^n + dtf.$$
2. Propose a method to solve the nonlinear Burgers equation :
$$-\Delta u + \nu u \partial_x u = 0.$$

1.2 2D Matrices

Considering a structured mesh of a 2D rectangle as shown in Figure, assemble the Mass and the Stiffness matrix.



Here is a way to proceed :

1. Write a function '[Ns,Coord,Nt,Vertex,Triangle,Index_vertex] = generatemesh(Mx,My,Lx,Ly)', where
   (a) Ns is the number of degree of freedom (i.e. the number of nodes)
   (b) Coord is a vertical list (with two columns) of the abscissa and ordinates of the nodes
   (c) Nt is the number of triangles
   (d) Vertex are the label of the nodes
   (e) Triangle is a 3 dimensional matrix : the last component is the label of the triangle, and Triangle(:, :,ℓ) is 2×3 matrix of the abscissa and ordinates of the nodes of the vertices of the ℓ-triangle.
   (f) Index_vertex is a $3\times$Nt matrix, $\ell-$column gives the three label of the vertices of the ℓ-triangle.
2. The local matrix is given by

```
function [m1,m2] = matrix_elementary(s)
% This function generates the elementary Mass matrix m1  and the elementary Rigidity
% matrix m2 for the P1-element method.
% s is a vector containing the (x,y)-coordinates of the three vertices of a given
%Triangle, meaning that s is of size 3x2.
% We remind that the area of a triangle can be computed from their coordinates (Xa,Ya),
%(Xb,Yb) and (Xc,Yc), the area S = 1/2*| det(Xb-Xa Xc-Xa;Yb-Ya Yc-Ya) |

  area_tri = 0.5*abs((s(2,1)-s(3,1))*(s(3,2)-s(1,2))-(s(2,2)-s(3,2))*(s(3,1)-s(1,1)));

%Mass Matrix

  m1 = area_tri/12*(ones(3)+eye(3));
  N = zeros(3,2);
  N(1,:) = s(3,:)-s(2,:);
  N(2,:) = s(1,:)-s(3,:);
  N(3,:) = s(2,:)-s(1,:);

%Stiffness Matrix

  m2 = N*N';
  m2 = m2/(4*area_tri);

end
```

   Here, m1 and m2 are the local contributions for the mass and stiffness matrices.
3. Write a function [Mass,Rigidity] = matrix_global(model) that generates the global Mass and Stiffness (also called *Rigidity* matrix) matrices.

2 With Free FEM

The purpose of this part of the T.P. is to become familiar with a finite element software, namely `FreeFEM++`. The documentation can be found here : `http://www.freefem.org/ff++/index.htm`

The syntax of this software is relatively simple. The work required consists of familiarization and then implementation. in slightly more complicated cases.

2.1 Getting started

1. To become familiar with the software, run some examples from the documentation.
2. Solve the problem of Dirichlet associated with a square domain from which an ellipse has been removed. Recall that this one is written as follows :
$$-\Delta u = 0,$$
for non-homogeneous Dirichlet Boundary Conditions on the boundary of the square, and non-homogeneous Neumann Boundary Conditions on the boundary of the ellipse.

2.2 More complicated examples

1. **Implementing Schwartz' Method :** in a square domain, implement a Schwarz method for the Laplace equation. More precisely : choose a rectangular domain, and define two sub-domains with an overlap area not reduced to a 1D interface. Then write the iteration between the two sub-domains.
2. **A time dependent example :** solve the equation
$$\partial_t u - \Delta u = 0,$$
keeping the space domain of Question 2 in the last section.
3. **Optimal Control :** See example 2.15.